An Introduction to Virtual Network Computing (VNC) for Connecting to NAS High-End Computers

Category: Productivity Hints

DRAFT

This article is being reviewed for completeness and technical accuracy.

Developed by ATT England, VNC provides a means to reduce X11 overhead on high-latency networks such as the Internet. In practical terms once a VNC session is underway latencies are on the order of seconds rather than minutes. VNC can make remote X11 applications useful instead of being tedious and non-productive.

The principle of operation involves a host server process (for example, Xvnc on Pleiades or Columbia at NAS) that communicates with X11 applications running on Pleiades or Columbia. The host server process translates the X11 images into something akin to an MPEG style video for display by a remote desktop/laptop's viewer client. Xvnc transmits images and image updates using a low overhead protocol to the user's viewer client.

Security and Firewalls

In the NAS environment VNC traffic is carried by a SSH tunnel much like SSH is used to tunnel X11 traffic. Using an SSH tunnel provides security because SSH encrypts tunnel traffic in both directions. If a user is already using SSH, then VNC traffic will find its way to/from NAS over current connections and through current firewalls. There is no need for any additional communication updates/authorizations.

Where is the VNC software?

The NAS Pleiades and Columbia systems are running with some versions of Linux. All the necessary VNC software is installed in */usr/X11R6/bin*.

On a user's desktop/laptop, there is no need to run a X11 server since in the VNC environment all the X11 work is done on the Pleiades front-end systems (pfe1-pfe12, bridge1-2) or the Columbia front-end (cfe2). However, a VNC client viewer is needed and it may have to be downloaded depending on whether this remote computer is running Linux, Mac or Windows. The client may already be installed in many Linux distributions and on recent versions of Mac OS X.

If you have a NAS supported desktop/laptop system, please note that:

- for NAS supported Linux workstations, a VNC client viewer (RealVNC version 4.1.2) should have been installed under /usr/bin/vncviewer
- for NAS supported Mac systems running the *Snow Leopard* OS, a VNC client called *Chicken of the VNC (version 2.0b4)* should have been installed under the */Applications* directory

If you have a Mac which is not supported by NAS, you can download *Chicken of the VNC (version 2.0b4)* from http://sourceforge.net/projects/cotvnc/

If you have a Windows desktop system, you can download free VNC clients from:

- http://www.realvnc.com/products/free/4.1/download.html
- http://www.tightvnc.com/download.php
- http://www.uvnc.com/download/index.html

Ask your local system administrator for help on installing the VNC client software.

Steps to a VNC session

The steps described below are not the only way in establishing a VNC session. However, they should prove to be more convenient in the sense that you do not have to manually find an available display number to use.

cfe2 is used below as an exmaple. You can substitute cfe2 with pfe1-12 or bridge1-2 if you want to establish a VNC session on a Pleiades front-end.

- **Step 1: SSH into cfe2**

  Starting a VNC connection/session is a matter of using SSH or some other SSH capable client to connect to cfe2. VNC is much easier to use if SSH Passthrough on your localhost has been set up.

  Note that in your *.ssh/config* file on your localhost, you do not need to enable SSH X11 forwarding, but you do need

  ```
  ForwardAgent yes
  ```

  Once SSH Passthrough is set up properly, you can establish a SSH connection from your localhost to cfe2.

  ```
  localhost% ssh cfe2
  cfe2%
  ```

- **Step 2: Invoke vncserver command on cfe2**

vncserver is a script that starts/stops/kills the actual VNC server Xvnc.

The first time you invoke *vncserver* on a server, you will be prompted to create a password for VNC. This password should be up to 8 characters in length. If you create a password longer than 8 characters, it will be truncated to the length of 8. This password is encrypted and saved in the *$HOME/.vnc/passwd* file on the server. Once this is done, you won't be prompted for a password on the server when invoking *vncserver* for subsequent VNC connections.

```
cfe2% vncserver -localhost

You will require a password to access your desktops.

Password:  <--- type in a password of your choice

Warning: password truncated to the length of 8.
Verify:  <-- retype your password

New 'X' desktop is cfe2:25

Creating default startup script /u/username/.vnc/xstartup
Starting applications specified in /u/username/.vnc/xstartup
Log file is /u/username/.vnc/cfe2:25.log
```

There are a few options to the *vncserver* command, such as *:display* (for setting the display number), *-geometry* (for setting the desktop width and height in pixel), etc. The **-localhost** option shown in the above example is a local security option that you should use all the time. It must appear as the last option or it won't get processed.

Similar to an X11 session, a VNC session uses a display number. If not supplied, the *vncserver* searches over the valid range from 0 to 99 and assigns the next free display number for your session. In the above example, a display number of 25 is assigned.

- **Step 3: Create a SSH tunnel from your localhost to the server**

The next step is to create a SSH tunnel from your localhost to the server. This is done by first escaping into an SSH sub-shell and specifying a local client's port number and a server's port number to use. The default SSH escape characters are <return> ~C (upper case 'C'). If you do not get the SSH prompt, repeat the <return>~C.

```
cfe2% ~C
ssh> -L 59xx:localhost:59xx
Forwarding port.
```

At the SSH prompt, provide a local client port and a remote server port. VNC by default uses TCP port 5900+xx. Thus, it is common to provide the value 59xx for both the local client port (the number before *localhost*) and server port (the number

after *localhost*). For example, the number 5925 can be used for both. If this number does not work, other numbers can be used, for example, 5825.

Note that the client port number and the server port number *need not* to be the same. Some may suggest using a very high client port number such as 22222 or 33333 since high port numbers are less likely to be reserved for other purposes. For example:

```
cfe2% ~C
ssh> -L 22222:localhost:5925
Forwarding port.
```

The maximum number allowed for the client port is 65535. Avoid using the local port numbers 0 - 1024 (root privilege required), 5900 (for Mac systems, reserved for some Apple remote desktop products), and 6000 - 6063 (reserved for local X window server). Use the *netstat -an* command to check what local port numbers have been used:

```
localhost% netstat -an | less
tcp46      0      0  *.5900                 *.*                    LISTEN
tcp4       0      0  *.22                   *.*                    LISTEN
```

The above example shows local ports 5900 and 22 are in use and should be avoided.

- **Step 4: Start the VNC viewer application on your localhost**

  - If your local host is a Mac and you have *Chicken of the VNC* installed, launch it. Open the Preferences panel from the "Chicken of the VNC" menu and select the Performance tab. Make sure the "Frontmost Connection" slider is not at its highest setting. If it is, move it down one notch. Close the Preferences panel. Now, open a new connection using the "New Connection" item from the "Connection" menu.

    In the popup window *Connect*, enter *localhost:22222* in the Host field (if your local port number is 22222 from step 3), and your VNC password in the Password field. Then click on the *Connect* button.
  - If your localhost is a Linux system, do:

    ```
    localhost% vncviewer localhost:localportnumber
    ```

    You should get a password prompt. Enter your VNC password that you created on the server.

    The localportnumber is the one you use in step 3. For example, if you choose 22222 as your local port, do:

```
localhost% vncviewer localhost:22222
```

If everything goes well, the Xvnc server will send a X11 window manager display to your localhost that will appear as an xterm in the viewer's window.

The default window manager is TWM, and there are a couple other window managers to choose from under */usr/X11R6/bin*, such as FVWM, MWM, etc. The KDE window manager, available under */opt/kde3/bin*, provides a GUI view of a user's files and includes a few useful tools. To use a non-default manager, for example KDE, modify your *$HOME/.vnc/xstartup* file on the host where your start *vncserver* as follows:

```
#twm &
/opt/kde3/bin/startkde &
```

Be aware that the KDE window manager needs more memory. For Pleiades users, it is recommended that you use the front-end nodes bridge1 and bridge2 instead of pfe1-8 if you want to use KDE for your VNC sessions.

You can also change the size and position of the xterm in your viewer's desktop by changing the values in the following line of the *$HOME/.vnc/xstartup* file on the host where you start *vncserver*. For example,

```
xterm -geometry 80x24+10+10 -ls -title "$VNCDESKTOP Desktop" &
```

specifies an xterm which is 80 characters wide, 24 characters high, at a position (10 pixels , 10 pixels) from the upper left corner of the VNC viewer's desktop.

The modifications to the *xstartup* file only take effect for a new VNC connection. You will need to stop the existing VNC server and start a new one.

The window manager's xterms is running on cfe2 itself. From this xterms, you can do tasks that you normally do on cfe2, for example, start an X application or ssh to other NAS systems. PBS jobs can also connect to a VNC session when the user provides the DISPLAY environment variable to the job. Specifically, in the xterm in the viwer's window, submit an interactive PBS job and set the DISPLAY variable to *vncserver_hostname:display_number* before starting an X application:

```
cfe2% qsub -I -lncpus=4,walltime=1:00:00
qsub: job 1030046.pbs1.nas.nasa.gov ready
PBS(4cpus)columbia21> setenv DISPLAY cfe2:25
PBS(4cpus)columbia21> xclock
```

- **Step 5: Shut down the server when you are done with the VNC session**:

  On each VNC server, there are a limited number of VNC sockets available. At the end of a session, be sure to exit the VNC application on your localhost so that others

can use the sockets. In the terminal window where you started up VNC, use the following command to clean up a few temporary socket files *vncserver* had created.

```
cfe2% vncserver -kill :xx (supply the original display number)
```

For example,

```
cfe2% vncserver -kill :25
Killing Xvnc process ID 3435054
```

DON'T manually kill vncserver. Doing so will leave lock and socket files (for example, /tmp/.X11-unix/X25, *$HOME/.vnc/cfe2:25.pid*, etc.) on the server.

Article ID: 257
Last updated: 04 Oct, 2011
The HEC Environment -> Your Environment -> Productivity Hints -> An Introduction to Virtual Network Computing (VNC) for Connecting to NAS High-End Computers
http://www.nas.nasa.gov/hecc/support/kb/entry/257/?ajax=1